Human Motion Analysis using Xsens MTw Awinda

Shahzeb Farruk, Hariprasad Munusamy, Palwasha Shaikh, *Member, IEEE,* and Qingyang Li.

***Abstract*—Motion capture systems play an essential role in health care, sport-training systems, gaming and animation. This paper conducts a human motion analysis using Euler angle data captured from a commercial body sensor network composed of Inertial-Magnetic Measurement Unit (IMMU) array called the Xsens MTw Awinda. The latter falls in the relatively new and popular category of the state-of-the-art micro-electromechanical system sensors-based motion capture systems that unlike their visual based counterparts are free from the constraints of sensitivity to the environment and its size and are cost-effective. The Xsens MTw Awinda body sensor network is used to collect motion data of the body key joints, and the data are delivered to the workstation wirelessly. Then, the proposed system uses analytical forward and inverse kinematics algorithms to analyze the Euler angle data while dealing with the issues of singularities and gimbal lock. Human body motion analysis system mainly involves the challenging tasks of designing the 3D human skeleton model and the development of a system that processes the sensor data referenced to a common base reference to be mapped with reference to other sensors based on a defined parent-child hierarchy relationship between the sensor nodes. As a result, the human body is abstracted into 16 key joints affecting human movement for the model. The prior is then transformed into a parent-child hierarchy with hip sensor node as the parent. The experimental results confirmed with 3 joints of the right arm that the model and the system are able to closely visualize the motion captured by the IMMU body sensor network. The proposed software solution works collaboratively with the hardware of motion capture system to provide good consistency of 3D skeleton stance and effective motion, and the implementation can be expanded to all 16-joints in the future for low-cost and simple analytical applications.**

***Index Terms*— Inertial sensors, Motion capture, Body Sensor Networks, Inverse Kinematics, Forward Kinematics, Human motion analysis, Direction cosine matrix.**

I. Introduction

Human motion capture and its analysis plays a vital role in many applications, such as animation, clinical, rehabilitation and healthcare analysis, human-computer interactions like virtual and augmented realities, sport training, gaming and many more [1]–[6]. Many motion capture (MoCap) tools and methods exist. One of the most widely used method is vision-based motion tracking [7]–[9]. This old yet standard technology involves using strategically placed markers on the human body that are tracked by optical tracking systems for MoCap. Despite its widespread use, the vision-based systems have many limitations [7], [8], [10]. Markers are susceptible to occlusion due to high mobility. Other drawbacks include cost, light sensitivity, and placement and difficulty in handling heavy equipment, which limits vision-based systems' applications to high-budget animation movies, gaming and clinical analysis.

A boom in low-cost, light and small sized body wearable sensors have made body sensor networks (BSNs) very popular. These body-worn systems can easily operate in obtrusive and small environments, as they are not dependent on an external motion capture device for recording the measurements [11], [12]. Typically, state-of-the-art micro-electromechanical system (MEMS) motion sensors, i.e. tri-axis accelerometers, gyroscopes, and magnetometers, fuse the data from all three types of sensors to provide orientation output that is highly accurate, reliable and drift free [7], [8]. The MEMS based MoCap system works extremely well for both continuous indoor and outdoor ambulatory motion measurements [5], [6], [9], [13]. A type of commercially available MEMS sensor is an Inertial-Magnetic Measurement Unit (IMMU) that comprises of a 3D linear accelerometer, 3D rate gyroscopes, 3D magnetometers and sometimes a barometer for correcting temperature errors in one package known as motion trackers. IMMUs can be purchased as single trackers or as part of BSNs, such as Xsens MVN, for full human MoCap [14].

There are two types of BSNs available for sale by Xsens MVN. The Xsens MVN Link [8] is one type that has the IMMU motion trackers all interconnected by wires in a suit, and requires a battery pack. The wired nature of the system ensures no data loss, but it is unsuitable for clinical and factory applications. As an alternative, in 2011 and later in 2016, XSens introduced MTw Awinda [15]. The chargeable and wireless motion trackers eliminate the need for cabling amongst the 16 body sensors as well as between the host and the tracker. Xsens developed a patented signal-processing pipeline and Awinda protocol [8], [16] to battle against wireless system's low data rate, data loss, and inter-tracker time synchronization issues. Xsens Kalman Filter (XKF-hm) is a sensor fusion algorithm specially developed for MTw Awinda to fuse all the sensor data together.

Wireless IMMU based BSNs are a very good choice for MoCap, and Xsens' MTw Awinda is chosen for conducting human motion analysis. The main objective is to record the raw orientation in terms of Euler representation (roll, pitch yaw) of all 16 sensors attached to different parts of the human body (head, shoulders, chest, arms, legs, foot, etc.),while performing some body motions. The individual orientation output of each IMMU is manipulated to form a hierarchical structure dependency of top to bottom or parent-child references. The processed data is visualized using a 3D skeletal model of the human body constructed according to the hierarchy, and human motion is analyzed in 3D.

Xsens has its own proprietary record and visualization tools called MVN Analyze [17] and MVN Animate [18], respectively, that takes the raw orientations of all 16 different sensors and shows the captured motion in 3D, and it could do

the latter in real-time as well. The proposed system aims to visualize motion capture data without the help of commercial Xsens visualizing system. The focus of this paper was to conduct an analysis of the human motion using the Xsens MTw Awinda system. The proposed system attempts at achieving the latter in a cost-effective and simplistic manner

The forward and inverse kinematics principles help to obtain the orientation in terms of Euler angles from the direction cosine matrix or rotation matrix and vice versa. Each of the sensor data which are with reference to the common reference frame called the East North UP, in this case, is mapped with reference to other sensors based on the parent-child hierarchy relationship between sensor nodes. Any system involving inverse kinematics like motion capture faces the singularity condition and the gimbal lock problem, which are explained in the Experiments section. The algorithm for processing and visualization works around these problems to achieve human motion in a 3D space. The processed data is supplied in arrays to the Unity3D software, a widely available visualization tool to visualize the captured human motion on a 3D skeleton.

The proposed system stands in a unique situation where it is a great and low-cost tool for someone that has the Xsens system, but cannot afford the MVN Analyze and MVN Animate tool or wants more low-level control for manipulation of raw Euler angle data of IMMU sensors. From an analyst's point of view the existing solution is compared with non-commercial, research-based systems that are built from low-cost IMMU sensors. It is highlighted in the next section how the proposed analytical solution is freed from concerns like calibration and data errors. Section III details the developed algorithm to achieve the outlined objective. The experimental set-ups are discussed in Section IV. Section V discusses the achieved results, and Section VI concludes this paper.

II. Literature Survey

A. Segmentation Procedures

The IMMU sensors attached to selected human body parts with suitable sensor fusion algorithms can give accurate and drift-free arbitrary orientation measurements. It is a well-established idea to use the raw orientation data of body segments along with the human biomechanical model. The recorded human motion can be replayed or recreated. The issue is that each sensor's coordinate frame is not aligned with the body coordinate frame. A transformation is required for the body segment measurements. In other words, the sensor coordinate frame needs to be aligned with the body coordinate frame, and a transformation matrix needs to be computed and applied. Between the IMMU sensors and the body parts, the body segment needs to be calibrated and aligned for meaningful and accurate results.

Reference [19] agrees that the human body lacks flat surface and rigid angles. Hence, it rejects the common assumption of pre-defined orientation and position of sensors for calibration. Instead, a simple three-posture calibration process is proposed. As the subject performs postures, the orientation measurements are collected for 17 body segments using 17 sensors placed on head, thorax, pelvis, shoulders, upper arms, forearms, hands, thighs, shanks and feet. Next, a segmentation process identifies and segments the postures using an average window algorithm, double-threshold technique and an intrinsic average algorithm to obtain a set of smoothed orientation measurements for each posture for each body segment. An iterative hand-eye calibration approach is proposed to finally get the sensor to segment calibration matrices. The prior method accesses or grades the quality of the calibration postures, so that the ungraded posture's effect on errors can be reduced. The entire process is validated on several human subjects that wear the IMMU BSN. The experiments confirmed that the calibration method is simple, easily performed by anyone, gives smoothed out orientation outputs, and is replicable

Reference [20] also proposes a sensor to segment calibration method like [19]. However, it requires an additional tool, which is a smartphone's compass or a commercial compass along with the IMMU sensors. Unlike [19], this calibration process only requires one posture where the subject simply faces the compass north and stands still to correct the IMMU's magnetometer. For improved accuracy in calibration, indoor and outdoor experiments revealed different constraints. Indoor environments pose the restriction of correcting the magnetometer by performing it near the body segment where the sensor is mounted. On the other hand, the calibration process outdoors can be performed in any position. The human motion is obtained by combining the vectors of each body segment during the capture of human motion. Experiments validated the accuracy of the calibration method using an optical measurement system called NDI Polaris Spectra System. The results of the pedestrian walking system with localization and posture capture application further verified the accuracy and proved the procedure simple and suitable for MoCap.

A skeleton calibration method for orientation measurement of the IMMUs is presented in [21] for both animals and humans. In this method, the limb postures are captured while the end-effectors, such as hands and feet, are matched with the known postures on the template, and a linear kinematic equation system of the skeleton dimensions is formed. However, the kinematic equation does not work well with single degree of freedom rotational joints inside the kinematic chain. So, the symmetric property of the skeleton is taken advantage of to make the dimension parameters identifiable. Experiments are conducted with IMMU sensors and a footprint template of humans to calibrate the lower limb dimension, and it is proved that this method controls dimension errors within 1 cm for humans for an accurate measurement than measuring limbs lengths alone.

In [20], the authors hope to implement the calibration process on the entire human body in the future, but in [19] a full body implementation is presented. The latter paper may have three different poses but is comparatively low-cost as it does not require an extra tool and is also independent of the environment of operation. Whereas in [20], by authors own admission the calibration method and algorithm may need improvements in accuracy and efficiency for full body MoCap. Compared with existing methods, the template based method of [21] is a quick and self-contained method which does not need extra measurement devices and require no specific posture positions for an accurate and centimeter level calibration of segments. Importantly, [21] unlike [19] and [20] works with any kind of skeleton including animal skeletons, and is also free

from asymmetric problems. Since the inertial MoCap systems are widely used nowadays, the method to generate an accurate model for precise behavior presentation is dependable on the application needs, system specific constraints, environment of operation, time, money and resources.

Xsens MTw Awinda BSN [7] used for this project frees us from the otherwise hectic task of calibration by employing Xsens proprietary algorithms and Awinda protocol. The users must use MT Manager or MVN Analyze tool and press "Configure" to calibrate all 16 sensors simultaneously while the subject wearing the sensors stands still with his arms down to the side. The sensor to segmentation process for all 16 joints is done by calculating rotation matrices for transformation, and forward and inverse kinematics is also used. The sensor to segment procedure was crucial to implement and conduct the visual analysis of the collected data. The accuracy of the simplistic solution is not very high, but the inputs are precise as Awinda protocol of the Xsens MTw [7] refines the data. It is subject to singularity and gimbal lock issues. However, the proposed software solution is very cost-effective. The latter issues of the proposed system are discussed in detail later in the paper. Table 1 compares our solution to other existing solutions in terms of calibration and sensor to link segmentation process from an analyst's point of view.

Table 1: Comparison of Calibration Processes and Segmentation Procedures.

Ref.	Calibration Process	No. of Sensors	Segmentation Procedure	Cost	Accuracy	Complexity
[19]	3 posture positions	17	Iterative-hand-eye collaboration procedure	Low	High	High
[20]	1 posture with a compass or smartphone	4	Combining the vectors of each body segment during MoCap	Med.	Low	High
[21]	-	17	Template based method	Low	High	Med.
Ours with [7]	1 posture and is automatically done with [7] using Awinda Protocol	16	Transformation matrices with forward and inverse kinematics	Low	Med.	Low

B. Data Errors and Sensor Fusion

MoCap data are typically represented as a time series in which each frame corresponds to a single pose usually represented by joint angles at a specific time instance. The collection of time-series poses is known as 3D skeleton motion data that are used for applications like action recognition, construction, animation, gaming, and virtual and augmented realities. But the raw motion data is subject to calibration errors, noise, and occlusion due to high mobility. Since the raw data given by most MoCap systems can't be used right away, the 3D motion data needs to be manipulated before applying it to the inputs of an application. Further, the human motion tracking data is seen as a multi-target tracking problem towards numerous body joints. The tough problem of accumulative errors and drift among multiple IMMU sensor data is also an issue, especially when they are fused together.

Reference [22] recognizes motion data refinement as an essential step in motion data preprocessing specially for low-cost and cheap sensors. Consequently, a new motion bidirectional recurrent autoencoder (BRA) based on bidirectional long short-term memory recurrent neural network (B-LSTM-RNN) for 3D skeleton human motion data refinement is presented. This method produces a manifold that allows for exploitation of both spatial and temporal relationships between present and past motion data. It learns statistical and kinematic information concurrently from noisy-clean training pairs and smoothens the data while abiding by bone length constraints. The processed data is of higher visual quality compared with state-of-the-art competitors and preserves the positional precision with low reproduction error.

Paper in [23] presents the design and experimental results of a quaternion-based Complementary Observer (CO) for human body motion tracking using IMMUs. The observer uses the attitude kinematic differential equation to describe the system model and pre-processes the sensor data using the Levenberg Marquardt algorithm (LMA). The quaternion calculated by the LMA is provided as input to the CO along with the angular rate data in this two-filter approach. The efficiency of the CO is experimentally proven through a factory robot and an IMMU during human segment motion exercises.

In [22], authors claim that the results demonstrate that the proposed method outperforms several state-of-the-art methods, is computationally fast, and works with all types of noise. But it has been only trained and tested for CMU human motion database and used simulated noise data. On the other hand, the method in [23] is a pre-processing data refinement step that can be applied to the raw MoCap data for higher positional accuracy between data with low reproduction. The latter unlike [22] is computationally expensive. Interestingly, both methods in [23] and [22] need a large set of training and testing data. Thus, making them both unsuitable for real-time refinement of acquired motion data.

The use of hybrid MoCap systems in clinical and factory environments are very popular. The precise localization of human operators in robotic workplaces is an important requirement to develop human-robot interaction tasks. Human tracking provides not only safety for human operators, but also context information for intelligent human-robot collaboration. Similarly, the accurate MoCap of humans in clinical studies can enhance the accuracy of the results and lead to better and more accurate diagnosis and solutions.

In [5], a hybrid tracking system for localizing a human operator precisely in a robotic workplace is developed. It consists of two components: an inertial MoCap system (GypsyGyro-18) and an Ultra-Wide Band (UWB) localization system (Ubisense). The inertial MoCap system records full body motions of the user's limbs, but these observed measurements are unable to give a true global position of the subject. To correct the errors in the global translational measurements, UWB is used for localization. A Kalman filter fusion algorithm is proposed that combines the global position measurements of both systems. Hence, the hybrid system takes advantage of both systems to provide high data rates and accurate limb orientations of inertial MoCap systems with accurate global user localization of UWB systems. The

accuracy of the system safeguards the operator and paves way for the development of intelligent interactive tasks with robots.

In [24], another hybrid system is proposed. It uniquely combines the inertial sensors with a Near-Infrared (NIR) optical sensor to record motion information accurately. Since the estimation of blood circulation changes in clinical investigations are affected by human mobility in loop systems, this wearable wireless NIR Spectroscopy (NIRS) cyber sensor system reduces the errors in the NIRS signals. It records the motion signals that consist of the highly accurate IMMU sensor data correlated with the optical sensor data to provide good estimation of artifacts in the NIRS signal. The motion fusion algorithm was applied to individual sensor signals to increase their accuracy by reducing their intrinsic limitations of imprecision and drifts. Thus, increasing the accuracy of the overall NIRS motion signals with improved SNR, and providing better detection of hemodynamic changes.

To compensate for the IMMU caused drift and accumulative errors, a multi-sensor system that combines IMMU data with time-of-arrival (TOA) is presented in [25]. On this basis, Cramér–Rao lower bound (CRLB) is derived in detail with consideration of both temporal and spatial related factors. Simulation results revealed that the proposed method has both spatial and temporal advantages, has significantly higher accuracy and less drift issues, compared with the traditional independent IMMU or TOA tracking methods. Furthermore, proposed method is verified in a 3D practical application scenario like walking.

Compared with state-of-the-art algorithms like Optimal Enhanced Kalman Filter (OEKF) and Zero velocity update (ZUPT) algorithm, proposed fusion method in [25] showed uniform and consistent performance, especially when moving direction changed. Without the requirements of external anchors, the method in [25] also has good stability and high tracking accuracy, thus it is suitable for wearable motion tracking applications.

In [5], the error in global local position of the user is counteracted with the UWB system. However, the latter hybrid system cannot handle spatial relationships between the robot and the skeleton of the operator. The motion fusion algorithm in [24] is used in the estimation and removal of the motion artifacts in the NIRS signal. The accuracy of the NIRS signal is very critical for the life-saving application of monitoring hemodynamics. Both the factory and clinical applications presented in [5] and [24] cannot afford errors in MoCap data. Hence, depending on the application one may use individual or combination of data refinement and sensor fusion techniques. The degree of accuracy required, available computational power, time, cost and needed depth and complexity will determine the choice of one technique over the other.

The Xsens MTw Awinda MoCap system [7] used in this project can process and refine the data using its proprietary MVN Analyze tool that employs the Awinda Protocol [8], [16]. The Awinda protocol is a patented wireless communication protocol constructed by a Strap Down Integration (SDI) algorithm to compute accurate angle measurements [16]. Further, the 16 sensors' data are filtered for errors and are fused together using Xsens's patented XKF-hm [16]. From an analyst's point of view, compared to other non-commercial systems built from low-cost IMU sensors, the Xsens MTw Awinda MoCap provides highly accurate data that is filtered for noise and errors. The data from this commercial system can be taken right away to perform visual motion analysis. Thus, saving time of the analyst by being very efficient and low in complexity. Table 2 shows the system used for analysis compared to existing systems in terms of data refinement and sensor fusion.

Table 2: Comparison of Data Refinement and Sensor Fusion Procedures to get Visualization Data.

Ref.	Raw Data Type	Data Refinement	Sensor Fusion	Effi-ciency	Com-plexity	Data Acc-uracy
[22]	Quaternion	BRA	-	High	Med.	High
[23]	Quaternion	CO with LMA	-	High	High	High
[5]	Quaternion	Using UWB measurements	KF fusion algorithm	Med.	Med.	High
[24]	Quaternion	Using NIR optical sensor measurements	Motion fusion algorithm	Med.	Med.	High
[25]	Quaternion	With TOA	CRLB	Med.	Med.	High
Ours with [7]	Both quaternion and Euler angles	MTw Awinda protocol	XKF-hm	High	Low	High

So, this project aims to analyze the human motion's raw orientation data captured by Xsens MTw Awinda and perform sensor to body segmentation using transformation to visualize the captured human motion. The details of the developed system to conduct the analysis are discussed in the following sections.

III. Algorithm

This section first details the issue of sensor frame references and how it is resolved. This is followed by describing the logic of forward and reverse kinematics, intrinsic rotations and the designed human skeleton hierarchy that is used to process the raw Euler angle data for visualization.

Each of the 16 IMMUs of Xsens MTw Awinda give orientations with respect to the East, North, Up (ENU) frame following a right-handed cartesian coordinate system. The ENU is used as the reference frame, and all the orientation values given by the sensors are in absolute space with respect to the ENU reference frame.

A. ENU Coordinates

The Local ENU cartesian coordinate system is formed by the plane tangent to the earth surface and fixed to a specific location. For many navigations, tracking and targeting applications, the ENU coordinate system, as shown in Figure 1, is preferred over Earth Centered Earth Fixed (ECEF) coordinate systems. The main advantage of ENU is that any scene construction can be quickly relocated by simply changing the geographic location corresponding to the scene's ENU origin. In sensors, the ENU is body-fixed to the device and is defined as the sensor coordinate system (S). The Figure 2 shows the ENU frame with x-axis, y-axis and z-axis.

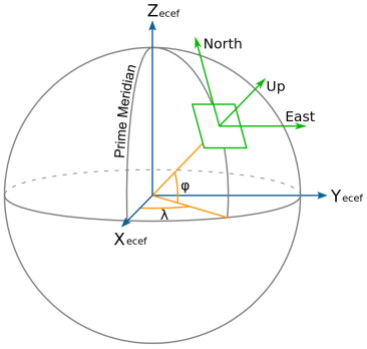

Figure 1: ENU coordinate system with respect to ECEF [26].

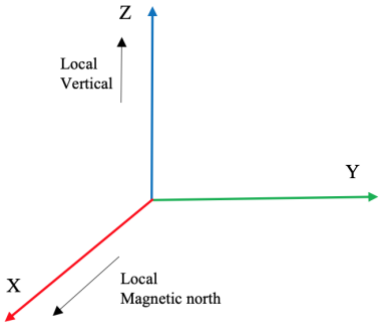

Figure 2: ENU Frame.

The IMMU sensor outputs the orientation, following the ENU coordinate system, as Euler angles and quaternions. The Euler angles describe the orientation of the rigid body with respect to a fixed coordinate system. Hence, the roll, pitch and yaw values describe the rotations of the frame in x-axis, y-axis and z-axis, and they are defined by (1), (2) and (3), respectively.

$$\varphi = Roll = rotation\ around\ X, [-180° \ to\ 180°] \quad (1)$$
$$\theta = Pitch = rotation\ around\ Y, [-90° \ to\ 90°] \quad (2)$$
$$\psi = Yaw = rotation\ around\ Z, [-180° \ to\ 180°] \quad (3)$$

B. *Forward Kinematics*

The rotation matrix or the direction cosine matrix (DCM) can be constructed either using the Euler angles or the quaternion values obtained from the sensors. Rotation matrix using Euler angles is given by (4), and the rotation matrix using quaternion values is defined in (5).

$$R = \begin{bmatrix} \cos(\theta)\cos(\psi) & \sin(\varphi)\sin(\theta)\cos(\psi) - \cos(\varphi)\sin(\psi) & \cos(\varphi)\sin(\theta)\cos(\psi) + \sin(\varphi)\sin(\psi) \\ \cos(\theta)\sin(\psi) & \sin(\varphi)\sin(\theta)\sin(\psi) + \cos(\varphi)\cos(\psi) & \cos(\varphi)\sin(\theta)\sin(\psi) - \sin(\varphi)\cos(\psi) \\ -\sin(\theta) & \sin(\varphi)\cos(\theta) & \cos(\varphi)\cos(\theta) \end{bmatrix} \quad (4)$$

$$R = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2q_1q_2 - 2q_0q_3 & 2q_0q_2 + 2q_1q_3 \\ 2q_0q_3 + 2q_1q_2 & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2q_2q_3 - 2q_0q_1 \\ 2q_1q_3 - 2q_0q_2 & 2q_1q_3 + 2q_0q_1 & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (5)$$

The Euler angles can be calculated either from the rotation matrix constructed using Euler angles or quaternions. Both (4) and (5) result in the same transformed Euler angle values, and their determinant is always 1, as defined by (6).

$$|R| = 1 \quad (6)$$

C. *Inverse Kinematic*

There is a possibility of ending up in a singularity condition while employing inverse kinematics. Equations (6), (7) and (8) are used to find the Euler angles while neglecting the singularity condition from DCM.

$$\Phi = \text{atan2}\left(, \frac{R_{33}}{\cos(\theta)}\right) \quad (6)$$
$$\Theta = \text{atan2}\left(-R_{31}, \sqrt{R_{32}^2 + R_{33}^2}\right) \quad (7)$$
$$\Psi = \text{atan2}\left(\frac{R_{21}}{\cos(\theta)}, \frac{R_{11}}{\cos(\theta)}\right) \quad (8)$$

The calculation of the rotation matrix, R, helps in performing the forward and inverse kinematics process, which helps calculate the position and pose of the joints. The sensors placed on the body are referred to as joints or nodes. It should be noted that there are 16 sensors attached to crucial body parts like hip, shoulder, leg, hands etc. to effectively capture the entire body motion, as can be seen in Figure 3 below.

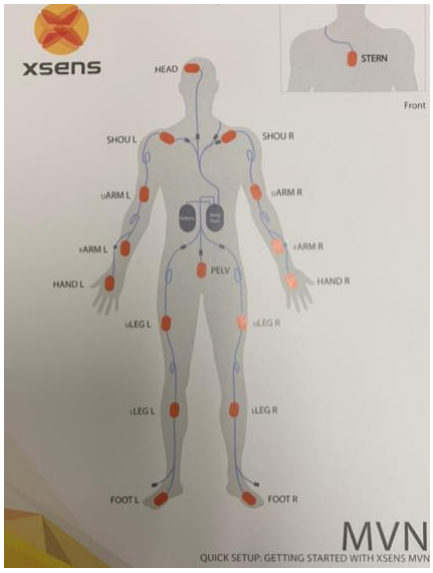

Figure 3: The image capture of the MVN quick-set up manual showing the ideal placement of the 16 wireless IMMUs of the Xsens MTw Awinda.

Using the accompanying MT Manager software tool of Xsens MTw Awinda, the data is obtained from the sensors. The wirelessly collected data defines each sensor's orientation in absolute space with respect to the ENU frame. The ENU frame is the base frame. When the links that are similar to the notion of body limbs are constructed by connecting at least two joints to construct a full human skeleton, the joints are no longer independent. Therefore, the rotation of every joint must be mapped to its parent joint based on the human skeleton hierarchy that is discussed later in this section.

So, the rotation matrix is calculated using the Euler values obtained from every sensor. The rotation or orientation of a joint can be mapped with respect to its parent joint using (9) derived from the properties of rotation matrices, where R_1^{ENU} is the rotation of joint 1 with respect to the ENU frame, R_2^{ENU} is the rotation of sensor or joint 2 with respect to the ENU frame, R_{ENU}^1 is the rotation of the ENU frame with respect to joint 1, and R_2^1 is the rotation of joint 2 with respect to joint 1.

$$R_2^1 = (R_1^{ENU})^T . R_2^{ENU} = R_{ENU}^1 . R_2^{ENU} \quad (9)$$

To match the initial frames of the joint with the reference ENU frame, we use intrinsic elementary rotations as defined by (10), (11) and (12). The $R_X(A), R_Y(B)$ *and* $R_Z(C)$ denote rotation about x-axis by an angle A, rotation about y-axis by an angle B , and rotation about z-axis by an angle C, respectively.

$$R_X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(A) & -sin(A) \\ 0 & sin(A) & cos(A) \end{bmatrix} \quad (10)$$

$$R_Y = \begin{bmatrix} \cos(B) & 0 & \sin(B) \\ 0 & 1 & 0 \\ -\sin(B) & 0 & cos(B) \end{bmatrix} \quad (11)$$

$$R_Z = \begin{bmatrix} \cos(C) & -\sin(C) & 0 \\ \sin(C) & \cos(C) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (12)$$

D. Intrinsic rotations

It is a rotation about a vector that is defined in the body frame or the moving frame. Body frames are frames that are attached to the joints. The rotation matrix for one or more intrinsic rotations is obtained by post-multiplying the individual rotation matrices in their respective order. Consider the example of rotation by 90° about y-axis followed by a rotation by 90° about the current z-axis as depicted in (13).

$$R = R_Y(90°).R_Z(90°) \quad (13)$$

E. Human Skeleton Hierarchy

The human skeleton is constructed in a hierarchical manner considering the hip as the root node. The skeleton hierarchy created can be seen in Figure 4. The hierarchical structure helps in creating links between the joints to form a human skeleton. When a parent joint makes a rotation, the corresponding child changes its position by moving along the link with the parent. The position of a child node depends on the orientation of the parent node. For example, when the Left Up leg makes a rotation and changes its orientation from its original state to the new state, the child joints Left low leg and Left Foot also move due to the changing position of their parent, Left Up leg. Likewise, when the Left low leg rotates, its child joint Left Foot also moves along the leg. Similarly, the rotation of the root joint, hip, will change the position of the entire body.

These joints or nodes in the hierarchy are rotated by pivoting the frames from the center of the joint to the end. The joint frames are moved from the center to the end of the joints based on the property of the link frames between the two joints. In the end, the rotation of the frame is preserved. So, the frame rotation remains the same throughout the link, and can be moved along the link, as shown in Figure 5.

F. Overall System

The overall process of the designed system begins with the placement of the 16 sensors on a subject's body as shown in Figure 3, and again in Figure 6 with the fixed initial frames for each sensor. The subject's body is then measured from sensor to sensor. Finally, the chosen MoCap system observes the joint movement and changes in orientation. These observations are wirelessly transferred to the workstation that has the MT Manager tool. The recorded data is then extractable in ASCII file format like .txt. The collected Euler angle data reflects the orientation with respect to the ENU frame. So, the observed changes in orientation are mapped with respect to the hip parent joint following the designed hierarchy of Figure 4. The prior is done by multiplying the rotation of a joint with the transpose of the rotation matrix of the parent joint. Using the inverse kinematics equation, the Euler angles are calculated from the resultant rotation matrix. Finally, the processed data is then ready to be fed to the virtualizing software, Unity3D, in this case. The Euler angles will define the x, y, z-axes rotation of every joint with frames pivoted at the end. Thus, the joint rotation will closely replicate and animate the captured human action for every time instance. The block diagram in Figure 7 summarizes the process involved from capturing the human motion to manipulating the acquired data and visualizing it successfully for human motion analysis.

- Hip
  - Left Up leg
    - Left low leg
      - Left Foot
  - Right Up leg
    - Right low leg
      - Left Foot
  - Left Shoulder
    - Left Upper arm
      - Left Lower arm
        - Left Hand
  - Right Shoulder
    - Right Upper arm
      - Right Lower arm
        - Right Hand
  - Head

Figure 4: The human skeleton hierarchy created based on the placement of the 16 IMMUs of the Xsens MTw Awinda..

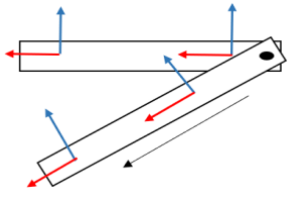

Figure 5: Frame movement across the joint link.

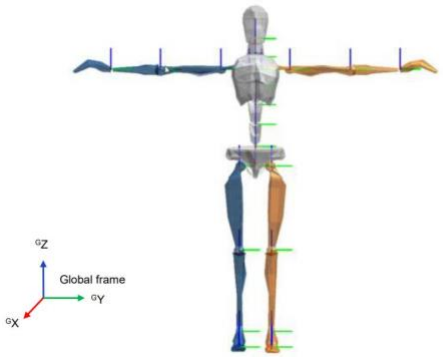

Figure 6: Sensor placement and their initial frames [16].

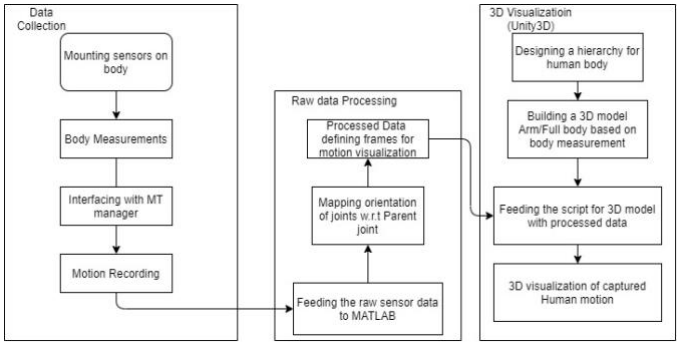

Figure 7:The block diagram of the proposed 3D human motion analysis system.

IV. EXPERIMENTS

This section begins with discussing the necessary steps of data collection and body measurements for conducting experimentation. Next, the required software tools and their purpose for conducting the experiments is highlighted. Finally, the implementation sub-section details each of the four experiments carried with each increasing in complexity to reach the desired objective of visualizing human motion for analysis.

A. Data Collection

The wireless Xsens MTW Awinda sensors have their own ENU reference frame, as shown in Figure 8. These sensors are mounted all over the body to capture the human subject's motion. The sensors are wireless. They use either Wi-Fi, Bluetooth or radio channel with the desired frequency and send their orientation outputs to the Awinda Master node. The frequency rate and available retransmission rate depends on the number of sensors connected to the master node as noted in Table 3.

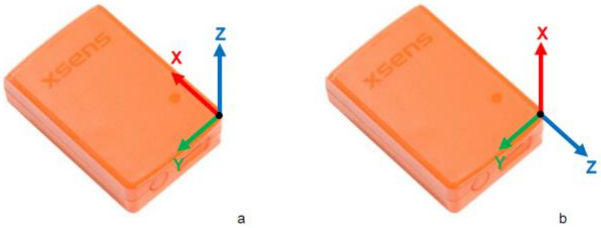

Figure 8: Sensors with ENU frame attached [16].

Table 3:Update rates and available slots for Xsens' MTw sensors [16].

Number of MTw Sensors	Update Rate [Hz]	Available retransmission slots
1-5	120	2
6-9	100	2
10	80	3
11-20	60	4
21-32	40	1

The Awinda USB dongle is used to interface the master with the MT Manager, which provides a graphical user interface (GUI) to collect the data from the sensors. MT Manager scans the port for sensors and configures them. The signal strength is an important factor, as the Awinda sensors must have stable power and be within the range of the Awinda Master to begin data collection. Figure 9 shows the successful connection and configuration of all 16 sensors in the MT Manager for recording the motion. The sensor movements are then captured wirelessly and stored on the workstation running the MT Manager. Once recorded, the data file can be imported into MT Manger for analysis as can be seen in Figure 9 with rotation of sensors in the MT Manager. Alternatively, data can be exported to the ASCII file format. The latter is done to extract raw sensor data for analysis. In the generated text file, the sensor nodes are identified by their unique ID that can be found at the back of the actual physical sensors. The sensor IDs of the MoCap system are then mapped to our designed human skeleton hierarchy for ease of reference as can be seen in Table 4. In total, four datasets have been recorded for analysis while making sure all the sensors are displaced from their initial orientations and positions

It should be noted that the sensors collect the data in packets at a sampling frequency dependent on the number of sensors. In our case, with 16 sensors the frequency rate was 60 Hz. The number of packets is dependent on the sampling frequency. These packets are sampled as frames by the MoCap system to be in synchronization with the motion video captured. For reference, a high-definition (HD) video captures 24 frames per second. This knowledge aided in the visualization process.

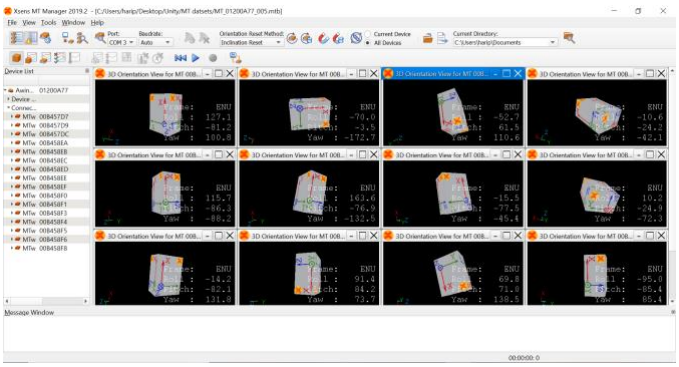

Figure 9:Data collection using MT Manager GUI.

Table 4: The mapping of joints names and Sensor IDs of the Xsens' MTw sensors with our designed human skeleton hierarchy.

Node ID	Joint Name	Sensor ID
1	Hip	00B458EE
2	Left upper leg	00B458F1
3	Left lower leg	00B458F8
4	Left foot	00B458F4
5	Right upper leg	00B457D7
6	Right lower leg	00B458EB
7	Right foot	00B458EF
8	Left shoulder	00B458EA
9	Left upper arm	00B458F3
10	Left lower arm	00B458F0
11	Left hand	00B458EC
12	Right shoulder	00B458F6
13	Right upper arm	00B457DC
14	Right lower arm	00B457D9
15	Right hand	00B458F5
16	Head	00B458ED

B. Subject's Body Measurements: Initial Body Position

Once the sensors are attached to the body of the subject, the subject's body needs to be measured from sensor to sensor. The measurement of the subject's initial body position is an

important step that can aid in building the skeleton in any chosen visualization platform. These body measurements help in identifying the initial position of the sensors on the body. Figure 10 shows the sensor placement on our subject's body following Figure 3. After initial body measurements have been recorded, the links connecting the sensors can be made with higher precision in terms of length and width.

Figure 10: Subject wearing the Xsens BSN.

C. Software Used

The MT Manager tool is an accompanying software of Xsens MTw Awinda that is used to record the 16-sensor data representing human motion. MATLAB is used to manipulate the acquired data using inverse and forward kinematics as well as DCM to achieve the orientation mappings following the designed hip originating parent-child hierarchy. Finally, the Unity3D tool is used to visualize the human motion by supplying the processed data. Blender is a popular animation tool that was used for initial attempts at visualization but was abandoned due to lack of low-level control. Table 5 below shows all the software platforms used to realize the human motion analysis.

Table 5: Software's used for processing

Software platform	Purpose
MT Manger	Motion capture using sensors
MATLAB	Performing orientation mapping and Euler angle calculations
Unity3D	3D visualization of the motion with the processed data
Blender	Initial trials for visualization

D. Implementation

Before performing any experimentation, the collected raw data had to be processed in MATLAB. The processed data is then used to conduct various visualization experimentations of increasing complexity.

1) Processing Raw Data in MATLAB

Based on the need of conducting an analysis on a particular body part like a single right arm, single right leg or the entire body, the sensor data is processed in MATLAB. The Euler angle data of the chosen set of sensor joints are given as input to the MATLAB code. The created MATLAB code takes the Euler angle's roll, pitch and yaw data and calculates the DCM of the joints for each packet of sensor data given following the algorithm logic discussed earlier. Then, the rotation of a joint is mapped with respect to its parent joint.

The hierarchy rotation matrix of all the joints mapped with its parent following Figure 4 were created in the MATLAB code, where (14) is the root node's rotation matrix notation and the ENU reference frame is denoted as G.

$$R_{hip}^{G} \tag{14}$$

Equations (15)-(18) are rotation matrix definitions of right arm's children where $R_{R\_shoul}^{hip}, R_{R\_uarm}^{R\_shoul}, R_{R\_uarm}^{R\_shoul}, R_{R\_larm}^{R\_uarm}$ *and* $R_{R\_hand}^{R\_larm}$ denote right shoulder, right upper arm, right lower arm and right hand, respectively. Similarly, (19)-(22) are rotation matrix definitions of left arm's children where $R_{L\_shoul}^{hip}, R_{L\_uarm}^{L\_shoul}, R_{L\_uarm}^{L\_shoul}, R_{L\_larm}^{L\_uarm}$ *and* $R_{L\_hand}^{L\_larm}$ denote left shoulder, left upper arm, left lower arm and left hand, respectively.

$$R_{R\_shoul}^{hip} = (R_{hip}^{G})^{T} * R_{(R\_shoul)}^{G} \tag{15}$$

$$R_{R\_uarm}^{R\_shoul} = (R_{R\_shoul}^{G})^{T} * R_{(R\_uarm)}^{G} \tag{16}$$

$$R_{R\_larm}^{R\_uarm} = \left(R_{R\_uarm}^{G}\right)^{T} * R_{(R\_larm)}^{G} \tag{17}$$

$$R_{R\_hand}^{R\_larm} = (R_{R\_larm}^{G})^{T} * R_{(R\_hand)}^{G} \tag{18}$$

$$R_{L\_shoul}^{hip} = (R_{hip}^{G})^{T} * R_{(L\_shoul)}^{G} \tag{19}$$

$$R_{L\_uarm}^{L\_shoul} = (R_{L\_shoul}^{G})^{T} * R_{(L\_uarm)}^{G} \tag{20}$$

$$R_{L\_larm}^{L\_uarm} = (R_{L\_uarm}^{G})^{T} * R_{(L\_larm)}^{G} \tag{21}$$

$$R_{L\_hand}^{L\_larm} = (R_{L\_larm}^{G})^{T} * R_{(L\_hand)}^{G} \tag{22}$$

Next, the rotation matrix definition of the head node was given by (23).

$$R_{head}^{hip} = (R_{hip}^{G})^{T} * R_{(head)}^{G} \tag{23}$$

Finally, the last two sets of body parts that are right and left leg were defined. Equations (24)-(26) are rotation matrix definitions of right leg's children where $R_{R\_uleg}^{hip}, R_{R\_lleg}^{R\_uleg}$, *and* $R_{R\_foot}^{R\_lleg}$ denote right upper leg, right lower leg, and right foot, respectively. Similarly, (27)-(29) are rotation matrix definitions of left arm's children where $R_{L\_uleg}^{hip}, R_{L\_lleg}^{L\_uleg}$ *and* $R_{L\_foot}^{L\_lleg}$ denote left shoulder, left upper arm, left lower arm and left hand, respectively.

$$R_{R\_uleg}^{hip} = (R_{hip}^{G})^{T} * R_{(R\_uleg)}^{G} \tag{24}$$

$$R_{R\_lleg}^{R\_uleg} = (R_{R\_uleg}^{G})^{T} * R_{(R\_lleg}^{G} \tag{25}$$

$$R_{R\_foot}^{R\_lleg} = (R_{R\_lleg}^{G})^{T} * R_{(R\_foot)}^{G} \tag{26}$$

$$R_{L\_uleg}^{hip} = (R_{hip}^{G})^{T} * R_{(L\_uleg)}^{G} \tag{27}$$

$$R_{L\_lleg}^{L\_uleg} = (R_{L\_uleg}^{G})^{T} * R_{(L\_lleg)}^{G} \tag{28}$$

$$R_{L\_foot}^{L\_lleg} = (R_{L\_lleg}^{G})^{T} * R_{(L\_foot)}^{G} \tag{29}$$

From the resultant rotation matrix of every joint, the Euler angles defining the joints rotation in x-axis, y-axis and z-axis are calculated. The resultant joint orientations are given with respect to the parent-child relationship mapping the rotation to the next immediate parent in the skeleton, refer Figure 4. The processed data for every packet is then fed as frames to the chosen visualization software.

2) Virtualization for Human Motion Analysis

Experiment 1: Demonstration of Forward Kinematics and Inverse Kinematics (using Euler angles)

This experiment demonstrates the forward kinematics and inverse kinematics, and also through exploration help identify several problems and their possible workarounds. The singularity and gimbal lock problems were identified and discussed as a result of this experimentation. The distinction between the two was necessary, since singularity and gimbal lock problems are often attributed to Euler angle usage [27]. Euler angle singularity is identified as purely mathematical in nature, and gimbal lock as a mechanical phenomenon. The next two subsections discuss the occurrence and resolution of the identified issues, and the last sub-section discusses the visualization experiment performed.

A. Singularity problem

In [28], singularity is looked at from a kinematics perspective. The rotation matrix in forward kinematic and inverse kinematics was calculated using Euler angles. The Φ, Θ, Ψ which are roll, pitch and yaw, respectively, are calculated provided only if $R_{23} \neq 0$ and $R_{33} \neq 0$ given by (30), (31) and (32).

$$\text{Roll} = \Phi = \text{atan2}(R_{32}/cos\theta, R_{33}/cos\theta) \quad (30)$$

$$\text{Pitch} = \Theta = \text{atan2}(-R_{31}, \pm\sqrt{R_{32}^2+R_{33}^2}) \quad (31)$$

$$\text{Yaw} = \Psi = \text{atan2}(R_{21}/cos\theta, R_{11}/cos\theta) \quad (32)$$

If $R_{23} = 0$ and $R_{33} = 0$ then singularity will occur. Since $\cos(\theta) = 0$, this will imply that $\theta = -90^\circ$ or 90°. For the case $\theta = 90^\circ$, we will have the R as defined in (33).

$$R = \begin{bmatrix} 0 & Sin(\varphi-\psi) & \text{Cos}(\varphi-\psi) \\ 0 & \text{Cos}(\varphi-\psi) & \text{Sin}(\psi-\varphi) \\ -1 & 0 & 0 \end{bmatrix} \quad (33)$$

For the case $\theta = -90^\circ$, we will have the R as defined in (34).

$$R = \begin{bmatrix} 0 & Sin(\varphi+\psi) & \text{Cos}(\varphi+\psi) \\ 0 & \text{Cos}(\varphi+\psi) & \text{Sin}(\psi+\varphi) \\ 1 & 0 & 0 \end{bmatrix} \quad (34)$$

Note, now we've two unknown values φ and ψ in R for a value of θ. If we now try to calculate φ and ψ by applying inverse kinematics, then $\varphi = \infty$ and $\psi = \infty$. So, there will be a problem in calculating the other two angles which are Φ and Ψ. As a result, in this condition both the angles will have infinite values, which is not a valid solution. In other words, the solution becomes meaningless. The practical implication with these values are dire. The motion of the joint will experience jerks, and its implementation on a physical manipulator would cause serious damage.

Through experimentation it was learnt that another expression for $\theta = -\text{asin}(R_{31})$ can be used, but it can cause its own set of problems. For example, for even a small error in the values of x in $asin(x)$ with x drifting from 0.99 radians to 1 radian there will be a relative error of 8.1°. Hence, to get better results $atan2(x)$ with one parameter is preferred that has less relative error. Though, $atan2(x)$ results in two possible values of θ.

Reference [28] presented Angle/Axis rotation in which rotation is about a vector with 4 parameters and it uses a rotation angle and a unit vector. But singularity is not eliminated with this method. Later, in [28] it was revealed that singularity doesn't exist if we use quaternions. So, it was learnt that quaternion values are preferable over Euler angle values for singularity condition to not exist.

B. Gimbal Lock Problem

In the gimbal lock problem, strange rotations occur between key frames unexpectedly. In Figure 11, each ring in the diagrams is called a gimbal. A gimbal can be defined as a pivoted support that allows the rotation of an object about a single axis. Reference [28] claimed that Euler rotations have a rotation order. In Figure 11, each of the axes are parented in the following order: Y to X, X to Z, and Z to the arrow. The Y-axis is top in hierarchy followed by X, Z and finally the arrow. When two of the gimbals match up, the rotation loses one degree of freedom. In Figure 11(b), x and z-axes line up or become coplanar hence the arrow can only rotate in one direction even if the rotation is applied to any one of the axes. To avoid gimbal lock, one may change the rotation order. But, still the gimbal lock problem will persist for any order of rotation. Reference [29] advocated using quaternions to avoid the gimbal lock problem.

Again, it was discovered that quaternion values are preferable over Euler angles. So, to not encounter the popular issues of singularities and gimbal lock quaternion values were used.

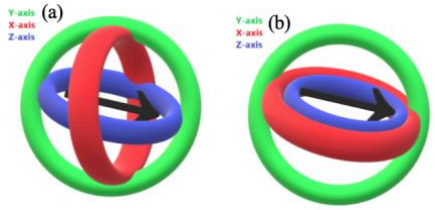

Figure 11: (a) The 3 gimbals without gimbal lock where X, Y, Z gimbals are represented in red, green, blue along with arrows, and (b) shows the occurrence of gimbal lock where X, Z are aligned [29].

C. Visualization in Unity3D

In forward kinematics demonstration, the Euler angles were given as inputs, and the rotation matrices were calculated from the roll, pitch, yaw (RPY) values. In inverse kinematics, from the rotation matrix that was calculated, the Euler rotation was computed. The visualization was performed on Unity3D. The two animated rectangles in Unity3D are shown in Figure 12 where the one on the left is performing rotation with Euler angle data while the one on the left is performing the rotation using quaternions. Both the cylinders are rotated about the Y-axis by 90° and continuously

about Z by 90˚ per second. The rectangle on the left with Euler angle data performs a rotation about X of local axes instead of rotating about the Z-axis to which the constant rotation is being applied. The rectangle on the right with quaternion values performs correct rotations. So, it was successfully experimented and confirmed that in a gimbal lock condition the rectangle with Euler values will not perform the desired rotations, but the one with quaternion values will not face any issues.

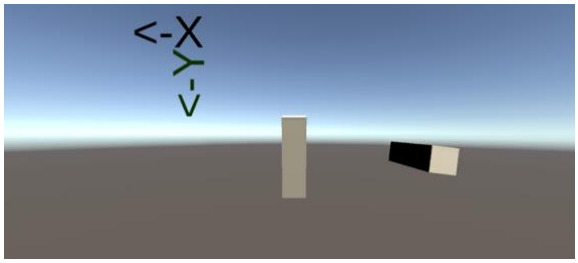

Figure 12: A snapshot from Unity 3D project named "gimbal lock" to demo gimbal lock in Euler and its absence in quaternions. Left rectangle is rotating using Euler values, and the right rectangle is rotating using quaternion values with both following the right-handed rotation system.

Experiment 2: Virtualization and demonstration of elementary rotation by using the data set captured for one sensor.

This experiment was done for exploring and understanding the raw data consisting of elementary motion captured by a single sensor. The data set is recorded for the right-hand sensor with ID: 00B458F5 using the MT Manager GUI as can be seen in Figure 13. The sensor's ENU was made to become close to zero rotation around the ENU frame as can be seen in Figure 14. In other words, the RPY was made to align values close to zero rotation. To visually demonstrate the elementary rotation about X, Y, Z-axes, a series of elementary rotations were first captured in sequence of: positive rotation about X, negative rotation about X, positive rotation about Y, negative rotation about Y, positive rotation about Z and negative rotation about Z.

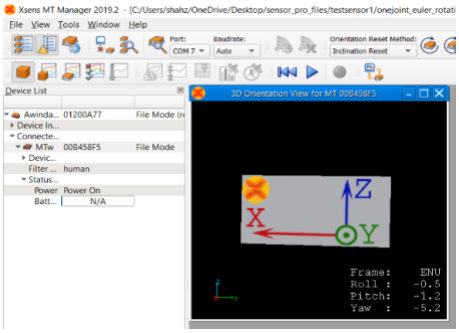

Figure 13: A screenshot from the MT Manager of a single sensor data set for elementary rotation.

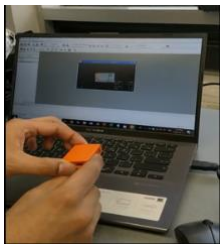

Figure 14: Moving a single Xsens MTw sensor for one sensor motion data collection.

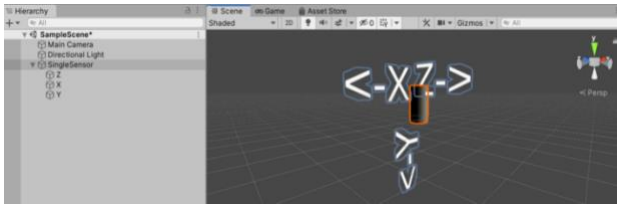

Figure 15: A screenshot in Scene Mode from Unity3D.

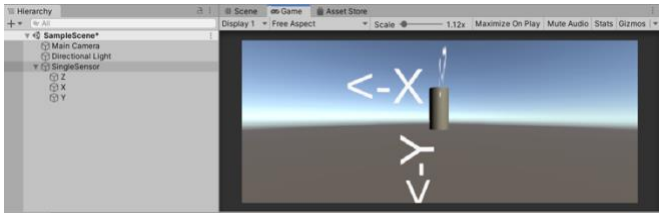

Figure 16: A screenshot in Game Mode from Unity3D.

It's easy to identify the ENU frame to which Xsens Awinda MTw sensors are calibrated in the physical world, but to measure magnetic North or East in the virtual environment like Unity 3D is absurd. Hence, we assumed that some frame in Unity3D is the local or rotating frame represented with text X','Y','Z' attached to the 'Single Sensor' Game-Object which is the 3D cylinder, as can be seen in Figure 15 and 16. The Game-Object's children is the initial frame or the base frame for the coordinate system in real-world, i.e. ENU coordinate frame. In Figure 15 and Figure 16, the Z- axis of rotation is out of the plane of the screen in Game Mode that is the compile and run mode in Unity3D. The Game Mode gives the view from the perspective of a fixed camera in the experiment, and the Scene Mode is more of an interactive tool analogous to programming or configuring a code before running it.

Differentiation between the local frame and the global frame in Unity 3D was learnt. It was also noted that the rotating frame in Unity3D engine of a Game-Object created is the local frame and not the global frame. Importantly, it was discovered that in Unity3D the frame rotations were Left-handed, but the data set recorded were with Right-handed rotations.

The singularities are not observed as the RPY angles are varied continuously in one of the axes in a single direction and not varied simultaneously by high values in all the axes together. Some variations exist in other axes but by very small amounts. Also, the gimbal lock problem is not witnessed as the rotations are close to elementary rotations, and the possibility of the axes getting aligned was very low.

Experiment:3 Visualization of 3-joint human right-arm motion in Unity 3D.

Building on the learnings and conclusions of Experiments 1 and 2, the raw RPY data set of 3 right-arm joints are given as input to our MATLAB code for processing.

Table 5:Right Arm sensors and sensor ID.

Node ID	Joint name	Sensor ID
1	Right upper arm	00B457DC
2	Right lower arm	00B457D9
3	Right hand	00B458F5

The right-arm sensor IDs are mapped to our node IDs and joint names as can be seen in Table 5. These sensors capture the motion of the right arm, and the observed raw sensor data defines the orientation of itself in absolute space. The 3-joint right arm datasets had 1,312 frames with 42 packets for every second. The MATLAB code helped in processing this data and map every joint's rotation with respect to the right-arm hierarchy, as can be seen in Figure 4. Equations (35)-(37) are rotation matrix definitions in our MATLAB code of right arm's children where $R^{G}_{Rpivot\_uarm}$, $R^{Rpivot\_uarm}_{Rpivot\_larm}$ *and* $R^{Rpivot\_larm}_{Rpivot\_hand}$ denote right upper arm root node, right lower arm and right hand, respectively. The ENU reference frame is denoted as G

$$R^{G}_{Rpivot\_uarm} \tag{35}$$

$$R^{Rpivot\_uarm}_{Rpivot\_larm} = \left(R^{G}_{Rpivot\_uarm}\right)^{T} * R^{G}_{Rpivot\_larm} \tag{36}$$

$$R^{Rpivot\_larm}_{Rpivot\_hand} = \left(R^{G}_{Rpivot\_larm}\right)^{T} * R^{G}_{Rpivot\_hand} \tag{37}$$

For the 3-joints modelled, the right upper arm is the root node. Right lower arm is the child of the upper arm, and the hand is the child of the lower arm and grandchild of the upper arm. Therefore, when the upper arm rotates and changes its orientation, the lower arm and hand moves along the link changing their position too. Likewise, orientation of the lower arm moves the hand. In order to achieve the latter, the rotation of the lower arm is mapped with respect to the upper arm, and the hand is mapped with respect to the lower arm. Thus, establishing the link relationship.

Using inverse kinematics, the Euler angle is calculated back from the rotation matrix for every joint. The process is repeated to find the joints' Euler angles for every packet instance. The processed data with the mapped orientation helps in visualizing the captured human right-arm motion. The MATLAB code's output describes the orientation of each joint, and defines the position and pose by joint link for every time instance. These processed Euler angles are now ready to be passed to the local frames of the pivoted joints in Unity3D.

It should be noted that the IMMUs were placed on the right-arm with the assumption that they were present at the pivots, since parallel axes have the same rotation. By default, in Unity 3D the axes of rotation are placed at the center of the Game-Object created. In this experiment, empty object pivots are created and placed as the reference frames of rotation for the model arms (labelled R_uarm, R_larm, R_hand in Unity3D) at the end of the link. Under these pivots the arms are nested as their children. The MATLAB output defining the orientation of upper arm, lower arm and hand are given to corresponding pivot points which is the end of the link. The parent child hierarchy of Unity3D code can be seen in Figure 17. Figure 18 shows the visualization of the right-arm motion in Unity3D.

- Rpivot_uarm
  - R_uarm
  - Rpivot_larm
    - R_larm
    - Rpivot_hand
      - Hand

Figure 17: The parent-child hierarchy of the right arm in Unity3D.

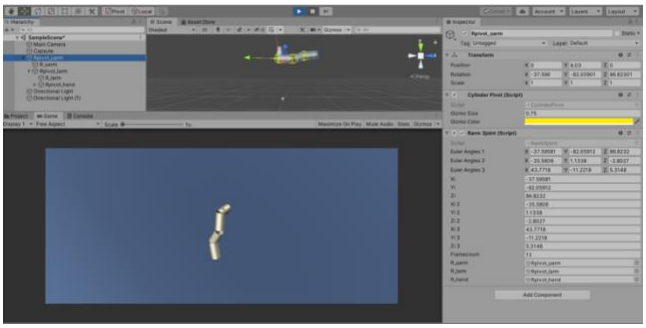

Figure 18: A snapshot from Unity 3D interface with the 3-joint arm model.

The comparison video for 3-joint model animated motion and captured motion was also created and the link can be found in the Appendix section. A series of snapshots from the recorded dataset in the real-world can be seen in Figure 19, and in Figure 20 the 3D model mapped rotations for the captured MoCap visualized successfully in Unity3D can be seen.

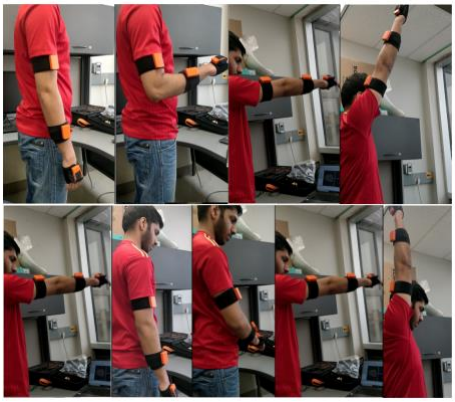

Figure 19: A Right arm motion capture in real-world using MTw sensors starting from raising the arm (top right to left) to lowering and raising the arm (bottom right to left).

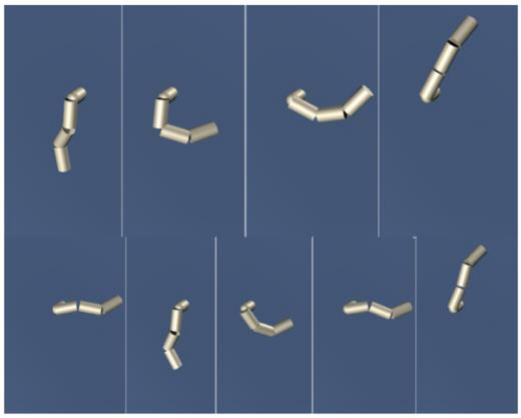

Figure 20: A Right-arm MoCap visualized in Unity3D using Euler angles implemented on a 3D pivoted model with 3-joint arm motion. It replicates the real-world motion of raising the arm (top right to left) to lowering and raising the arm (bottom right to left).

Experiment 4: Visualization of 16-senor Full human body motion in Unity3D

The 3D human skeleton hierarchy of Figure 4 was built in Unity 3D from scratch using 3D cylinder Game-Objects. The logic for the three moving joint arm is expanded to develop the full 3D human skeleton model. Figure 21 shows the built 3D skeleton with local frame attached to every joint in Unity3D.

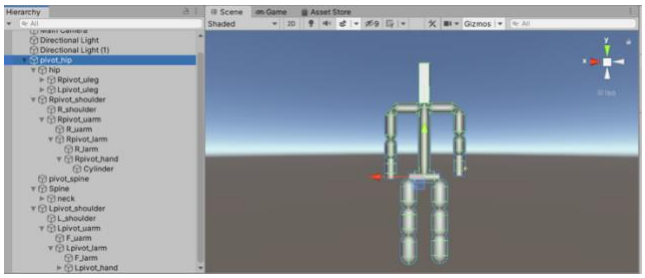

Figure 21: Full body skeleton built in Unity3D.

By default, in Unity3D the axes of rotation are placed at the center of the objects created. Like the last experiment, in this experiment empty object pivots are created and placed as the reference frames of rotation for the 3D model limbs at the end of the link. Sensor ID mappings used for this model can be found in Table 4. In the 3D model created, the pivot_hip is the base parent joint, unlike Experiment 3 in which the base was simply the right shoulder. So, the dependencies will change and therefore the mapping. The rotation matrices definitions with reference to hip as the parent were defined in (15)-(29).

It is noticed that the processing time increases for processing 16 sensor raw Euler values, and the alignment of each of the local frames becomes a daunting task. Additionally, the singularity and gimbal lock are observed. Due to limited computation power and time-constraints, the 16 sensor raw data could not be processed and fed for visualization with quaternion values. However, all the groundwork is laid for future experimentation and exploration.

So, it can be concluded that the same logic and algorithm of right-arm can be easily scaled up to 16-sensor full body MoCap visualization in Unity3D. And, the use of quaternions values will probably enhance the accuracy of the visualized motion to closely match the captured human motion. Further, the work can be expanded to process and visualize the MoCap data in real-time.

V. RESULTS

The results of the captured motions' visualization are evaluated by comparing the real-life MoCap video of the subject with its 3D visualization in Unity3D. To accurately and objectively evaluate the aim of human motion analysis and visualization, the MoCap video of our human subject was synchronized with the recreated motion in Unity3D. The sampling frequency and video frame rate were carefully matched to empirically evaluate the output.

Figure 22 shows the result of the Experiment 2's visualization of a single sensor or joint recorded data sets in Unity3D. The latter experiment is evaluated by doing a visual side-by-side comparison of the real-time MoCap's matched video with simulation in the Unity3D platform. It is noticed that the object in Unity3D rotates according to the passed values from the real sensor in the given axis. Realization of a single joint did not need any MATLAB processing for data. There was no frame mapping or hierarchy involved. Thus, easing the process of visualization for a single joint on the visualization platform.

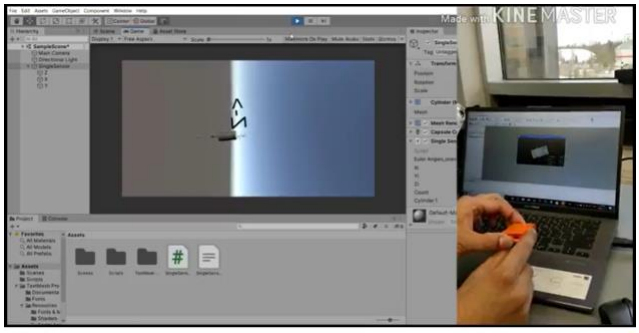

Figure 22: Single joint visualization results compared with real-life MoCap video for evaluation.

Realization of a right-arm motion in Experiment 3 was more challenging. The joints were linked, which created dependencies and resulted in more data for processing. The increase in data proportionally increased the computational complexity. However, in comparison to other existing systems, this system is still of low complexity with its usage of simple algorithms. The hierarchy relation was achieved by performing data processing with the raw Euler angle data using MATLAB. Figure 23 shows the resulting visualization of motion in Unity3D being compared with the real-life and real-time MoCap video of the right-arm motion. It is concluded from Experiment 3, that the same algorithm and approach can be expanded to achieve the full body Mocap's visualization.

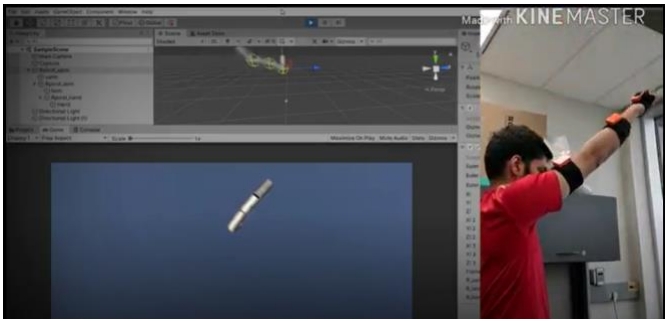

Figure 23:Results of Right arm motion visualization.

Finally, in Experiment 4 learning from the results and evaluation of Experiments 2 and 3, the full 3D human skeleton was created, as can be seen in Figure 21, according to the recorded body measurements of the subject and the defined hierarchy in Figure 4. The model visualization shows a close motion to that of the real-world recorded dataset. However, there are a few discrepancies observed in the motion as a combined result of gimbal lock problem and singularities present in Euler rotations applied. So, the results were less accurate, as Euler angle values were used. Difficulties in 16-sensor raw data processing, and frame rotation of joints in the skeleton were also encountered, especially in terms of time. As mentioned earlier, using quaternion values can improve the accuracy in the future of the full body motion visualization.

Table 6 compares the results of the experiments conducted. Experiment 1 was not part of the human MoCap, so it is not included in the comparison. It was noted that the packet count depends on the sampling frequency of the sensors. The packets count and capture duration does not influence the complexity. The time spent in data processing and visualization increases

with increase in the number of joints in the human body MoCap process. The visualization results were relatively more accurate in Experiment 2 and 3 compared to Experiment 4. In comparison to surveyed systems, our system is less accurate. Since, a quick and simplistic approach is demonstrated for a low-cost and low-complexity human motion visual analysis for an analyst.

Table 6: Comparison of experimental result (Exp. = Experiment).

Exp.	Body part	No. of Joints	No. of Sensors	Packet Count	MoCap Duration	Relative Visualization Accuracy
2	N/A	1	1	6230	20s	High
3	Right Arm	3	3	1312	31s	High
4	Full Body	16	16	2711	65s	Low

The major challenge in visualizing the captured human motion was without a doubt the existence of singularity and gimbal lock issues that locked the joints and gave improper rotation at some time instances. Another obstacle encountered was figuring out the creation of dependencies between the joints and defining a hierarchy that can accurately model the human skeleton. Formulating the mappings of the rotation of the joints from global to body segment and forming a parent-child relationship was also quite a difficult task. It was also observed that the time for processing raw data increased with increase in the number of joints. This is due to the increasing number of computations for processing the large amount of sensor data and mapping every joint according to the hierarchy relationship. The simplicity and low-complexity implementation of this system for quick motion analysis can be considered a trade-off for the latter.

So, the successful visualization of one joint and the right-arm motion are considered a breakthrough in implementing the human motion analysis using Xsens MTw Awinda's IMMUs. To reach the latter achievement, many incremental experiments were conducted. Experiment 1 led to the identification of the gimbal lock difficulty and the singularity issue in inverse kinematics. The prior led to the realization that better and more accurate motion can be visualized using quaternions for data processing. For an accurate full body motion visualization both offline and in real-time, quaternion value inputs are recommended in the future.

The video links showing the results of all the experiments conducted can be found in the Appendix section.

VI. Conclusion

A novel low-cost and viable system for quick human motion analysis of MEMS based MoCap system's data is proposed. The system is unique in its purpose. It is a software system that works with the Xsens MTw Awinda's refined yet raw Euler angle data of 16 individual sensors. The task of analyzing human motion is challenging and a popular topic of research. This paper attempts at approaching this problem in a simplistic and scalable manner by utilizing the inverse and forward kinematics algorithm with DCM. The sensor data was successfully transformed from the common ENU reference frames to sensor data referencing other sensors using the carefully defined hip originating parent-child hierarchy. Processing the huge amount of data for even a minute long of MoCap is computationally challenging, but software tools like MATLAB proved helpful.

Popular animation tools like Blender with its .bvh file format did not provide enough low-level control to animate the 3D human skeleton. The frame rate and the animation tool's common reference frame definitions were not in line with our desired requirements and led to abnormal motion. MATLAB was another alternative that was explored and abandoned, as it did not meet our simplicity objective. Finally, Unity3D with its accurate physics engine proved to be the best choice.

Visualization aspect of this paper was filled with trials and errors. It was understood through experimentation that using Euler angles brought on issues of singularities and gimbal lock. So, using quaternions for the inverse kinematics was demonstrated as the better choice. The right-arm motion with 3 joints experimentally verified the designed system. It is proved that this low-cost and computationally simple system helps visualize captured motion with good consistency. In the future, the implementation can be easily scaled to accurately demo all 16-joints for low-cost and simple analytical applications.

Contributions

Shahzeb Farruk helped the whole team co-ordinate and have been a consistent team player since day one, providing insights and ideas for the project and ways to debug, test all issues. Wrote, analyzed and presented in-depth understanding of singularities with the code in MATLAB concluding the practical implications. Experimented and presented immense understanding of gimbal lock problem along with code and demonstration in Unity 3D. Along with doing experiments, codes for single sensor motion, followed by 3 IMMU joint arm motion, and also 16 IMMU sensor 3D model in Unity3D apart from constructing 3D models and projects for all of them. Moreover, Initiated and collaborated on idea for 3 joint arm logic and algorithm, also worked on creating the logic, algorithm and project for 16 IMMU sensor 3D model. Wrote and conducted Experiment:1,2, and 4. And in experiment 3 did whole Unity part including coding, logic. Also, proofread this document and help create the presentation, and created videos for results. In addition, worked on bvh format MoCap on Blender as well, which was abandoned because of inconsistent behavior of reference frames making it incompatible with the datasets from Xsens Awinda MTw.

Hariprasad Munusamy started with exploring 3D visualization platforms. He worked on the project implementation algorithm and skeleton hierarchy, solved the frames mapping issues using elementary rotations. He came up with a MATLAB process and logic for manipulating the raw data and mapping the orientation of joint with respect to the parent joint. He created 3D models and human skeleton in Unity3D. He collaborated with teammates to work on the tests and experiments collectively. He worked on C# code for Unity3D, and contributed in report by writing Algorithm, some parts of Experiments and Results. He did proof study and peer

review on groupmates work, and did many trial runs in Blender platform using .bvh and other file formats, initially.

PALWASHA SHAIKH conducted a wide literature survey of calibration and segmentation procedures, data refinement issues and sensor fusion relating to human motion analysis, especially from an analyst's point of view. She compared existing systems with the proposed overall system composed of the XSens MTw Awinda hardware and our simple and low-complexity software solution. Also, she wrote the Abstract, Index Terms, Introduction, Literature Survey, References and Conclusion sections, and rewrote, edited and completed the Algorithm, Experiment and Results sections of this paper. She created and designed the PowerPoint Slides with Li completing the last two sections of the slides. She diligently worked on maintaining the overall IEEE formatting details and project requirements, and proof-read this paper. She achieved the first successful attempt of human motion analysis on Unity3D with the asset of MVN 3D human Skeleton of 16-joints, and wrote successfully a C# code to run the raw Euler angle data without any transformations for a single right-arm motion visualization. Later, she expanded the C# code to operate all 16 joints with raw Euler angle data without any processing, but the unprocessed data of 2000 frames led to abnormal motion due to frame referencing issues. Prior issues were resolved in collaboration with Hariprasad and Farruk.

Qingyang LI was passion on the project when she get the assignment. She works with teammates at computer lab for 4 to 6 hours. During the data collection process, she wears the equipment carefully and get good image and data collection for the project beginning. For the implementation part, during the discussion meeting, she was assigned to work on MATLAB part. She builds a 6 DOE machine arm although it is not mentioned in the report but the rotation and algorithms are applied in the future project work. She gives ideas for the project according to her very beginning machine arm Simulink file. During the discussion in COVID-19 period, her teammates and she always talk few hours for the project and keep trying to solve the problem in the MATLAB part. For the Untiy3D part, she gathers the solution and combine them in C# and create the coding files for 3D modelling. For the paper working part, she works on the Introduction, Literature Survey, and Algorithm equations and explanation. In the end, she creates the presentation.

APPENDIX

Results and code are uploaded in below drive link:
https://drive.google.com/drive/folders/18R9MUnMy_8ZOVPhlEFFycdllivCU04Av

YouTube Video Link:
Visualization of 3-joint human right arm motion in Unity 3D:
https://youtu.be/zA1tWpJPCAA

Single Sensor Elementary rotation using Euler Rotations in Unity 3D:
https://youtu.be/7psPbk4-_Xo

Gimbal Lock Problem using Unity 3D:
https://youtu.be/AvYjiupEAEc

Experiment-1, RPY.m MATLAB code's execution video:
https://youtu.be/PkzSiASsCto

ACKNOWLEDGMENT

We would like to thank Dr. Mohamed Atia for his precious and humble guidance even during this very hard time of the COVID-19 pandemic. Last but not the least, we would also like to thank the academic group who helped us fix the sensor issue for data collection in a very short span of time.

REFERENCES

[1] G. Zizzo and L. Ren, "Position tracking during human walking using an integrated wearable sensing system," *Sensors (Switzerland)*, vol. 17, no. 12, p. 2866, Dec. 2017, doi: 10.3390/s17122866.

[2] X. Li, S. H. Han, M. Gül, and M. Al-Hussein, "Automated post-3D visualization ergonomic analysis system for rapid workplace design in modular construction," *Autom. Constr.*, vol. 98, pp. 160–174, Feb. 2019, doi: 10.1016/j.autcon.2018.11.012.

[3] K. Kim and Y. K. Cho, "Effective inertial sensor quantity and locations on a body for deep learning-based worker's motion recognition," *Autom. Constr.*, vol. 113, p. 103126, May 2020, doi: 10.1016/j.autcon.2020.103126.

[4] L. V. Ojeda, P. G. Adamczyk, J. R. Rebula, L. V. Nyquist, D. M. Strasburg, and N. B. Alexander, "Reconstruction of body motion during self-reported losses of balance in community-dwelling older adults," *Med. Eng. Phys.*, vol. 64, pp. 86–92, Feb. 2019, doi: 10.1016/j.medengphy.2018.12.008.

[5] J. A. Corrales, F. A. Candelas, and F. Torres, "Hybrid tracking of human operators using IMU/UWB data fusion by a Kalman filter," in *HRI 2008 - Proceedings of the 3rd ACM/IEEE International Conference on Human-Robot Interaction: Living with Robots*, 2008, pp. 193–200, doi: 10.1145/1349822.1349848.

[6] Y. Zhang, Y. Fei, L. Xu, and G. Sun, "Micro-IMU-based motion tracking system for virtual training," in *Chinese Control Conference, CCC*, 2015, vol. 2015-Septe, pp. 7753–7758, doi: 10.1109/ChiCC.2015.7260871.

[7] G. Bellusci, F. Dijkstra, and P. Slycke, "Xsens MTw : Miniature Wireless Inertial Motion Tracker for Highly Accurate 3D Kinematic Applications," *Xsens Technol.*, no. April, pp. 1–9, 2013.

[8] M. Schepers, M. Giuberti, and G. Bellusci, "Xsens MVN : Consistent Tracking of Human Motion Using Inertial Sensing," *Xsens Technol.*, no. March, pp. 1–8, 2018, doi: 10.13140/RG.2.2.22099.07205.

[9] T. Huynh-The, C. H. Hua, T. T. Ngo, and D. S. Kim, "Image representation of pose-transition feature for 3D skeleton-based action recognition," *Inf. Sci. (Ny).*, vol. 513, pp. 112–126, Mar. 2020, doi: 10.1016/j.ins.2019.10.047.

[10] T. B. Moeslund, A. Hilton, and V. Krüger, "A survey of advances in vision-based human motion capture and analysis," *Computer Vision and Image Understanding*, vol. 104, no. 2-3 SPEC. ISS. Academic Press, pp. 90–126, 01-Nov-2006, doi: 10.1016/j.cviu.2006.08.002.

[11] C. Xu, J. He, X. Zhang, C. Yao, and P. H. Tseng, "Geometrical kinematic modeling on human motion using method of multi-sensor fusion," *Inf. Fusion*, vol. 41, pp. 243–254, May 2018, doi: 10.1016/j.inffus.2017.09.014.

[12] R. Gravina, P. Alinia, H. Ghasemzadeh, and G. Fortino, "Multi-sensor fusion in body sensor networks: State-of-the-art and research challenges," *Inf. Fusion*, vol. 35, pp. 1339–1351, May 2017, doi: 10.1016/j.inffus.2016.09.005.

[13] S. Bakhshi, M. H. Mahoor, and B. S. Davidson, "Development of a body joint angle measurement system using IMU sensors," in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 2011, pp. 6923–6926, doi: 10.1109/IEMBS.2011.6091743.

[14] Xsens Technologies, "Motion Capture." [Online]. Available: https://www.xsens.com/products. [Accessed: 23-Mar-2020].

[15] Xsens Technologies, "MTw Awinda." [Online]. Available: https://www.xsens.com/products/mtw-awinda. [Accessed: 23-Mar-

2020].
[16] Xsens Technologies, “MTw Awinda User Manual.” Xsens North America, Inc., Culver City, USA, 2018.
[17] Xsens Technologies, “MVN Analyze.” [Online]. Available: https://www.xsens.com/products/mvn-analyze. [Accessed: 23-Mar-2020].
[18] Xsens Technologies, “MVN Animate.” [Online]. Available: https://www.xsens.com/products/mvn-animate. [Accessed: 23-Mar-2020].
[19] Y. T. Liu, Y. A. Zhang, and M. Zeng, “Sensor to segment calibration for magnetic and inertial sensor based motion capture systems,” *Meas. J. Int. Meas. Confed.*, vol. 142, pp. 1–9, Aug. 2019, doi: 10.1016/j.measurement.2019.03.048.
[20] N. Choe, H. Zhao, S. Qiu, and Y. So, “A sensor-to-segment calibration method for motion capture system based on low cost MIMU,” *Meas. J. Int. Meas. Confed.*, vol. 131, pp. 490–500, Jan. 2019, doi: 10.1016/j.measurement.2018.07.078.
[21] Q. Yuan, I. M. Chen, and A. W. Sin, “Method to calibrate the skeleton model using orientation sensors,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2013, pp. 5297–5302, doi: 10.1109/ICRA.2013.6631335.
[22] S. Li, Y. Zhou, H. Zhu, W. Xie, Y. Zhao, and X. Liu, “Bidirectional recurrent autoencoder for 3D skeleton motion data refinement,” *Comput. Graph.*, vol. 81, pp. 92–103, Jun. 2019, doi: 10.1016/j.cag.2019.03.010.
[23] H. Fourati, N. Manamanni, L. Afilal, and Y. Handrich, “Complementary observer for body segments motion capturing by inertial and magnetic sensors,” *IEEE/ASME Trans. Mechatronics*, vol. 19, no. 1, pp. 149–157, Feb. 2014, doi: 10.1109/TMECH.2012.2225151.
[24] M. R. Siddiquee *et al.*, “Sensor Fusion in Human Cyber Sensor System for Motion Artifact Removal from NIRS Signal,” in *International Conference on Human System Interaction, HSI*, 2019, vol. 2019-June, pp. 192–196, doi: 10.1109/HSI47298.2019.8942617.
[25] C. Xu, J. He, X. Zhang, X. Zhou, and S. Duan, “Towards human motion tracking: Multi-sensory IMU/TOA fusion method and fundamental limits,” *Electron.*, vol. 8, no. 2, Feb. 2019, doi: 10.3390/electronics8020142.
[26] Mike1024, “File:ECEF ENU Longitude Latitude relationships.svg - Wikimedia Commons,” *Wikimedia Commons*, 2017. [Online]. Available: https://commons.wikimedia.org/wiki/File:ECEF_ENU_Longitude_Latitude_relationships.svg. [Accessed: 11-Apr-2020].
[27] E. G. Hemingway and O. M. O’Reilly, “Perspectives on Euler angle singularities, gimbal lock, and the orthogonality of applied forces and applied moments,” *Multibody Syst. Dyn.*, vol. 44, no. 1, pp. 31–56, Sep. 2018, doi: 10.1007/s11044-018-9620-0.
[28] B. Sciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*, 1st ed. London: Springer-Verlag London, 2009.
[29] Y. Yao, Z. Du, X. Huang, and R. Li, “Derivation and simulation verification of the relationship between world coordinates and local coordinates under virtual reality engine,” *Virtual Real.*, pp. 1–7, Aug. 2019, doi: 10.1007/s10055-019-00397-7.

Palwasha Shaikh (M’16) earned the BEng. in computer engineering with summa cum laude from the American University of Kuwait (AUK), Salmiya, Kuwait in 2019. She is currently pursuing the degree of MSc. in electrical and computer engineering at University of Ottawa (UO), Ottawa, ON, Canada.

From 2015-19 she was a Teaching Assistant with the Tutoring Center (TRC) at AUK. From 2016-19, she was a Research Assistant at AUK with the Department of Electrical and Computer Engineering. Currently, she is a Research Assistant at UO with Dr. Hussein T. Mouftah. She has worked on and published conference and transaction papers in the field of FPGAs, embedded systems, traffic signal control problems, light-fidelity, IoT and connected vehicles in the past. Her current research interests include smart connected and autonomous electric vehicles, smart grids, and charging infrastructure and schemes for connected and autonomous electric vehicles.

She has received IEEE Kuwait Section’s Award for Appreciation in 2018 and 2019 for conducting lectures on topics like IoT and embedded systems.